Context Switching

Ein Vokabular-Präfix für FORTH-Bibliotheken

mehr Klarheit im Denken

besser strukturierte Quelltexte

übersichtlichere Benutzerschnittstelle

Manfred Mahlow Forth-Tagung 2017

(Kontakt: Vorname PUNKT Nachname AT forth-ev.de)

Wir verwenden Präfixe in Wortnamen aber wir implementieren sie nicht:

ohne Vokabular Präfix	**mit** Vokabular Präfix
RS485_PORT	RS485 PORT
rs485-read	rs485 read
1w.get	1w get
port:hi	port hi
i2c.init	i2c init
@i2c.ee	i2c eeprom @
timer0.init	timer0 init

Beispiel 1 : FM-Tuner mit Color-LCD-Display

Wörterbuch ohne Vokabular Präfixe (430eForth)

```
WORDS
 FM-Radio+RDS FMT_APP FMT_INIT FMT? FMT-timer ?up/dn FMT_FREQ- FMT_FREQ+
FMT_VOL- FMT_VOL+ FMT_SEEK_DN FMT_SEEK_UP FMT_RDS_CLR RDS? RDS_RESP_CLR RDS_PSN?
RDS_PSN_CLEAR RDS_PSN R() FM_RDS_STATUS FM_RDS_INIT FMT_PSN_CLR FMT_STATUS
FMT_VOL FMT_RSQ FMT_FREQ lcd.r lcd.type.r FMT_LCD_MASK grey __Si4735-FM__
FM_TUNED_FREQ_STRING FM_TUNED_FREQ FM_AGC_STATUS FM_RSQ_STATUS FM_TUNE_STATUS
FM_STEREO/MONO FM_STEREO FM_MONO FM_VOL- FM_VOL+ FM_SET_VOL FM_GET_VOL
FM_SEEK_DN FM_SEEK_UP FM_SEEK_START FM_TUNE FM_DOWN FM_UP __Si4735__ Si4735_?CTS
Si4735_?TUNED Si4735_SET_PROPERTY Si4735_GET_PROPERTY Si4735_GET_INT_STATUS
Si4735_POWER_UP Si4735_POWER_DOWN Si4735_io __I2C__ i2c?addr i2c?ready I2CRDTO
i2c.ready i2c?busy i2c.read i2c.in i2c.write i2c.out i2c?ack i2c.rdy i2c.ack
i2c.nak i2c.rx i2c.nak? i2c.tx i2c.stop i2c.start i2c.init RSHIFT NIP __ST7735__
lcd.cr lcd.type lcd.spaces lcd.emit >5x7 5x7-font ALIGN ALIGNED C, IC!
lcd.write.5x7 lcd.write.1x7 lcd.clr lcd.init lcd.normal lcd.bg lcd.fg
lcd.txt.size p/d dy dx lcd.page yellow blue green red white black >rgb
lcd.fill.rect lcd.write.pixels lcd.pix/rect lcd.set.rect lcd.init lcd.RGB565
lcd.LANDSCAPE lcd.PORTRAIT lcd.format lcd.COLMOD lcd.MADCTL lcd.RAMWR lcd.RASET
lcd.CASET lcd.DISPON lcd.DISPOFF lcd.SLPOUT lcd.SLPIN lcd.SWRESET lcd.cmd.d
lcd.cmd.c lcd.cmd lcd.write.cmd lcd.D/C- lcd.D/C+ lcd.CS- lcd.CS+ spi.write.word
spi.write.byte spi.init LSHIFT 4DUP 2OVER ms end-code code WITHIN 2SWAP -ROT
COLD APP! HI WORDS .S DUMP ' VARIABLE CONSTANT CREATE : ] ; ." $" ABORT" WHILE
```

Beispiel 2 : FM-Tuner mit Color-LCD-Display

Wörterbuch mit Vokabular Präfixen (noForth V)

```
@)words
 FM-RADIO  RADIO-TEST  SI4735.FS  SI4735  I2C-CORE.FS  I2C  LOCAL
 ST7735-LCD.FS  >5X7  5X7-FONT  LCD  SPI  VOC  COMPILE,  4DUP  TOOLS\
 DUMP  .S  NOFORTH\  BN  DM  HX  MS  POSTPONE  [']  DOES>  IMMEDIATE
```

Auf die Bibliotheken/Kontexte kann man gezielt zugreifen:

```
@)FM-RADIO words
 APP  INIT  .STATUS  TIMER  ?UP/DN  VOL-  VOL+  PS-  PS+  LCD  RDS (B)
 '  ..  WORDS  DEFINITIONS (4) OK.0
```

```
@)FM-RADIO LCD words
 PSN_CLR  .STATUS  .VOL  .RSQ  .FREQ  MASK (6)
```

```
@)FM-RADIO RDS words
 RDS?  PSN?  PSN_CLR  PSN  RESP_CLR  RESP()  STATUS  INIT (8)
```

@)SI4735 words
FM ?CTS ?TUNED SET_PROPERTY GET_PROPERTY GET_INT_STATUS POWER_UP
POWER_DOWN IO (9)

@)SI4735 FM words
TUNED_FREQ_STRING TUNED_FREQ AGC_STATUS RSQ_STATUS TUNE_STATUS
STEREO/MONO STEREO MONO VOL- VOL+ SET_VOL GET_VOL SEEK_DN
SEEK_UP SEEK_START TUNE DOWN UP (12)

@)i2c words
?ACK ACK NAK RX NAK? TX STOP START INIT (9)

@)LCD words
WHITE .R TYPE.R CR TYPE EMIT WRITE.5X7 WRITE.1X7 CLR INIT
NORMAL BG FG TXT.SIZE P/D DY DX PAGE YELLOW BLACK WHITE BLUE
GREEN RED >RGB FILL.RECT WRITE.PIXELS PIX/RECT SET.RECT INIT
RGB565 LANDSCAPE PORTRAIT FORMAT COLMOD MADCTL RAMWR RASET CASET
DISPON DISPOFF SLPOUT SLPIN SWRESET CMD.D CMD.C CMD WRITE.CMD
D/C- D/C+ CS- CS+ (34)

@)SPI words
WRITE.WORD WRITE.BYTE INIT (3)

Implementierung (für noForth V)

- Ein Vokabular Präfix ist ein Immediate Wort mit einer eigenen Wortliste. Es liest ein Wort aus dem Eingabestrom und sucht es in seiner Wortliste.

- Wird das Wort gefunden wird das Execution Token in Abhängigkeit von der Variablen STATE ausgeführt oder compiliert.

- Wird das Wort nicht gefunden, gibt es eine Fehlermeldung.

- Ist der Eingabestrom leer, wird eine Eingabeaufforderung angezeigt und auf eine Eingabe gewartet.

```
\ Usage:

\ voc name           \ creates a new vocabulary prefix

\ name definitions   \ makes name (names wordlist) the current
                     \ compilation context

\ Example:

voc i2c  i2c definitions

  variable sid      \ slave id

  : start ... ;     \ start an I2C Bus transmission

  : stop  ... ;     \ stop an I2C Bus transmission

    ...             \ more I2C definitions

forth definitions
```

```
\ i2c words    \ shows all i2c words and stays in the i2c context
\              \ entering a name or .. closes the context
\
\ i2c start    \ executes/compiles the word start from context i2c
\
\ i2c ' start  \ returns the execution token of the word start
```

```
: voc ( "name" -- )
  \ Create a vocabulary prefix. Generic version, context names are not
  \ displayed with ORDER and SEE.
    wordlist create , immediate
    does> ( a -- )
          \ Set the current vocabulary prefix context and execute dovpx.
          @ to voc-context dovpx ;
```

```
: dovpx ( "name" -- )
  \ Read the next word from the input stream, find it in the actual voc context
  \ and execute (state=0 or word=immediate) or compile it (state<>0).
    search-voc-context dup 0= ?abort
    state @ = if compile, else execute then ;
```

```
: search-voc-context ( "name" -- 0 | xt 1 | xt -1 )
  \ Find name in the actual voc context.
    bl-word count ( a u ) \ bl-word waits for input
    2dup voc-context search-wordlist ( adr len wid -- 0 | xt 1 | xt -1 )
    ?dup
    if
      2swap 2drop
    else
      voc-root search-wordlist ( adr len wid -- 0 | xt 1 | xt -1 )
    then ;

wordlist constant voc-root ( -- wid ) \ the root wordlist visible in all vocs

value voc-context ( -- wid )  \ points to the wordlist of the last used voc
```

Der komplette Quelltext für noForth-v und AmForth findet sich im Wiki der Forth-Gesellschaft im Ordner templates im e4thcom-0.6.1.tar.gz Archiv.